

Laboratory 6 - Estimation, filtering and system identification - Prof. M. Taragna

Exercise: model identification for a water heater, using nonlinear models (real data)

Contents

- [Introduction](#)
- [Procedure](#)
- [Setup](#)
- [Neural Network model class](#)
- [Choosing the best model](#)

Introduction

The program code may be splitted in sections using the characters "%%". Each section can run separately with the command "Run Section" (in the Editor toolbar, just to the right of the "Run" button). You can do the same thing by highlighting the code you want to run and by using the button function 9 (F9). This way, you can run only the desired section of your code, saving your time. This script can be considered as a reference example.

```
clear all, close all
```

Procedure

1. Load the .mat file containing input-output data
2. Remove the mean value from the data
3. Plot input-output data
4. For each input/output delay nk from 1 to 5, and for each order na=nb (NNARX) from 1 to 5
5. Identify the NNARX model using the estimation dataset
6. Compute the predicted output on the validation dataset
7. Compute RMSE and MDL criteria in order to select the most suitable model

Setup

```
% Step 1: load of data

load heater

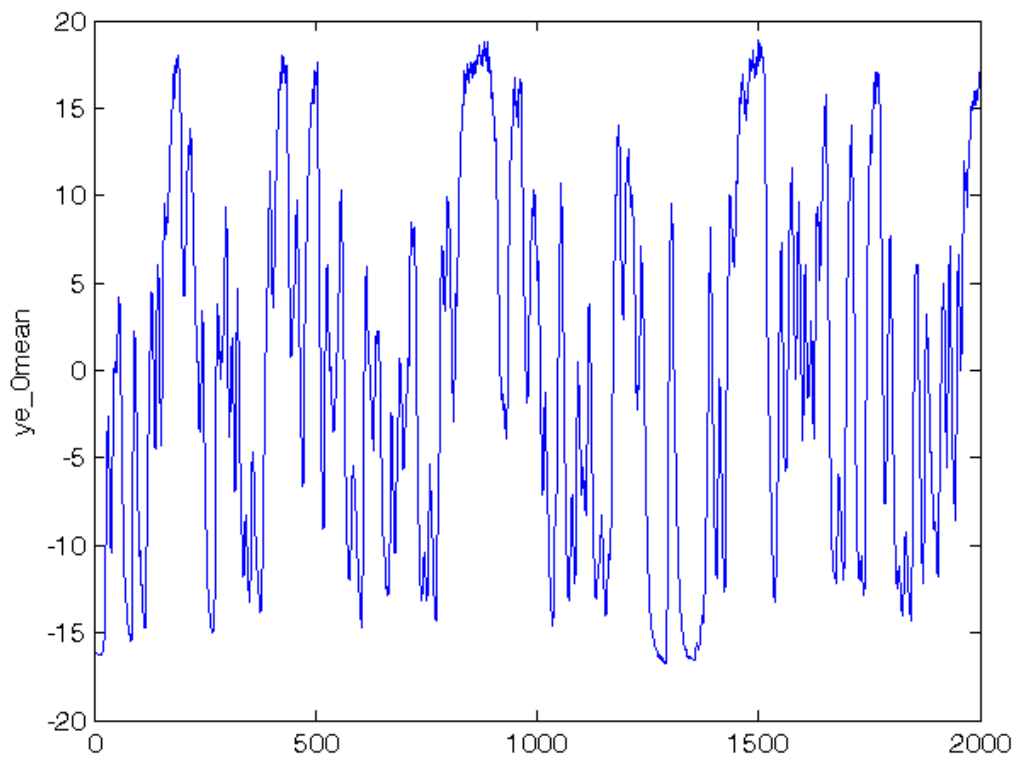
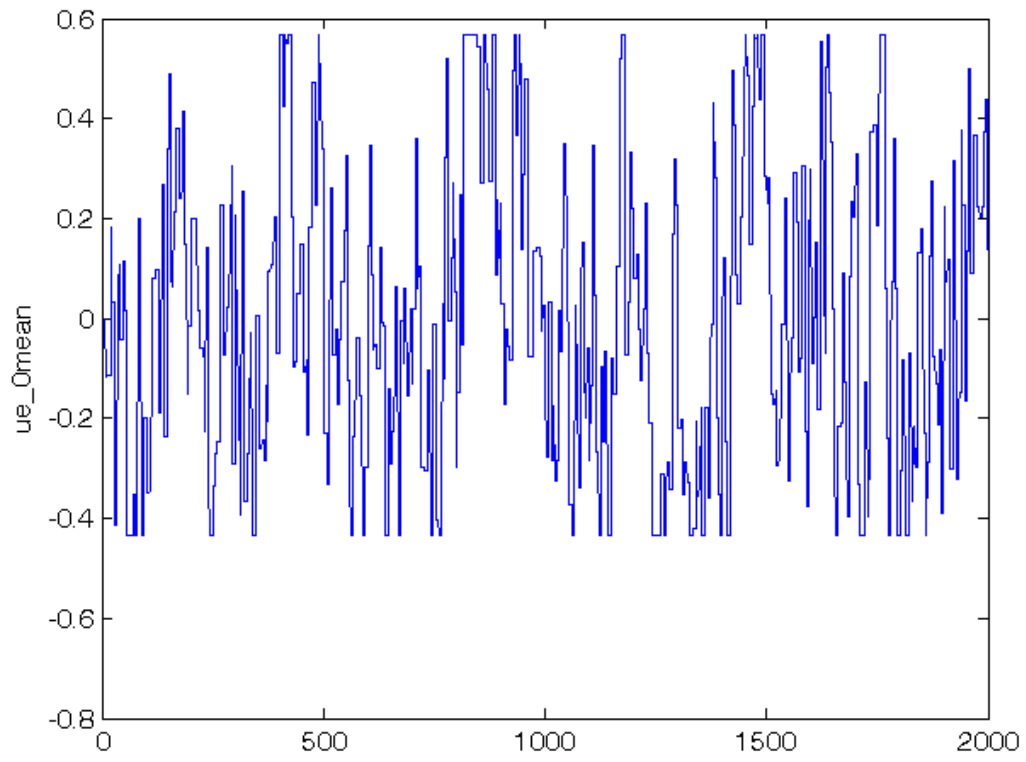
% Step 2: remove the mean value

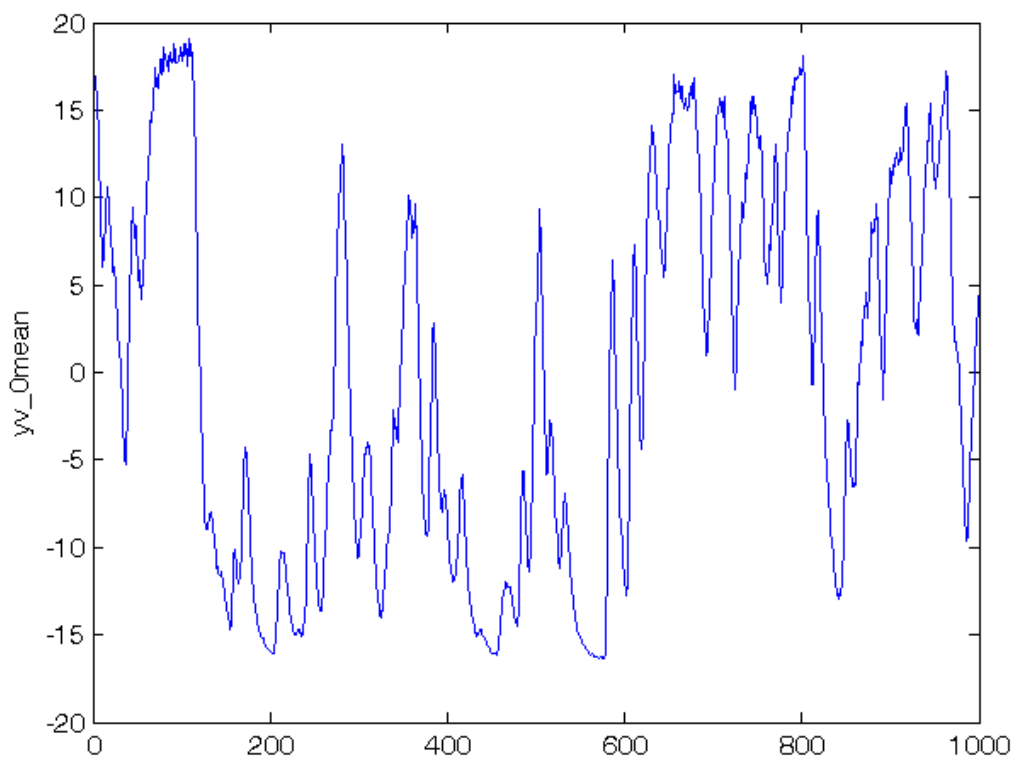
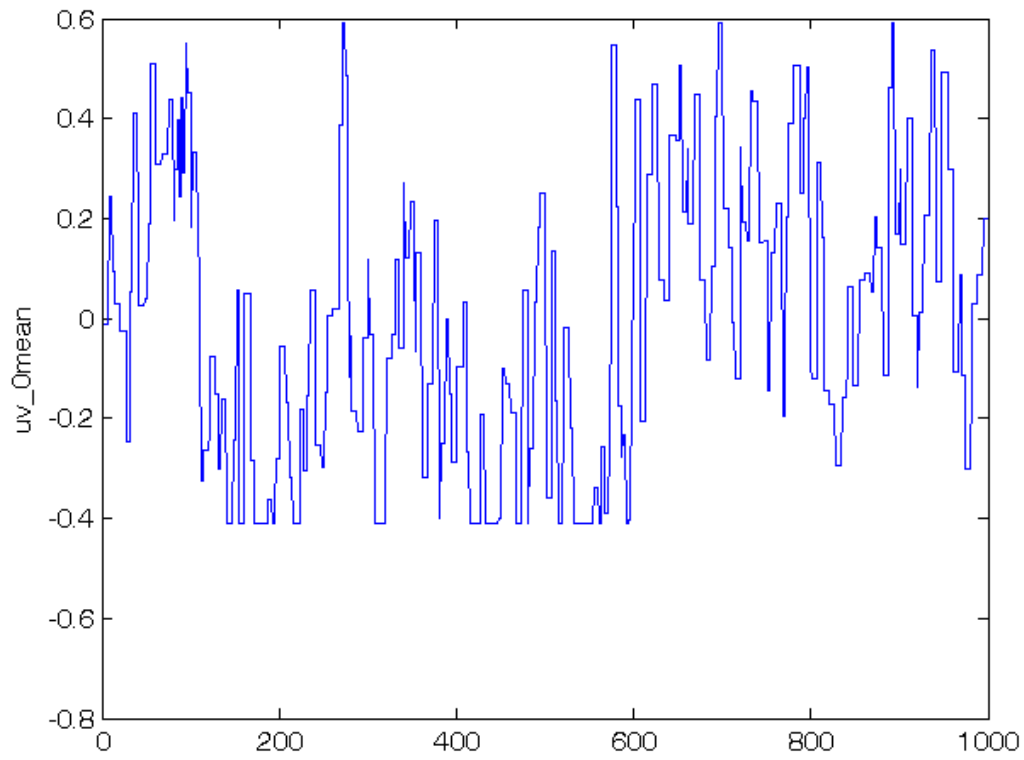
ue_0mean = ue - mean(ue);
ye_0mean = ye - mean(ye);
uv_0mean = uv - mean(uv);
yv_0mean = yv - mean(yv);

% Step 3: plot of input-output data

Ne = length(ue); % number of estimation data
Nv = length(uv); % number of validation data

figure, plot(ue_0mean), ylabel('ue\_0mean'),
figure, plot(ye_0mean), ylabel('ye\_0mean'),
figure, plot(uv_0mean), ylabel('uv\_0mean'),
figure, plot(yv_0mean), ylabel('yv\_0mean')
```





Neural Network model class

Define the model structure of the neural network having:

- $r \geq 5$ hyperbolic tangent neurons (or nodes or units) in the hidden layer
- 1 linear neuron (or node or unit) in the output layer

```
Net.Def=[ 'HHHHHH' ; 'L-----' ] ;
```

```

% For any Neural Network ARX model NNARX(na,nb,nk), define its orders:
% - na = # of past measurements used in the regressor phi
% - nb = # of past inputs used in the regressor phi
% - nk = minimum input-output delay

% Step 4: definition of NNARX parameters

na=1;
nb=na;
nk=1;

```

Set the maximum number of iterations to 500

```
trparms=settrain; trparms=settrain(trparms,'maxiter',500);
```

Perform the NNARX identification, using

- as output signal Y, a row vector with $\text{dim}(Y) = [1 * (\# \text{ of data})]$
- as input signal U, a matrix with $\text{dim}(U) = [(\# \text{ of inputs}) * (\# \text{ of data})]$ In the SISO case, both Y and U are row vectors

```

% Step 5: estimation of NNARX parameters

[W1,W2]=nnarx(NetDef, [na,nb,nk], [], [], trparms, ye_0mean', ue_0mean');

```

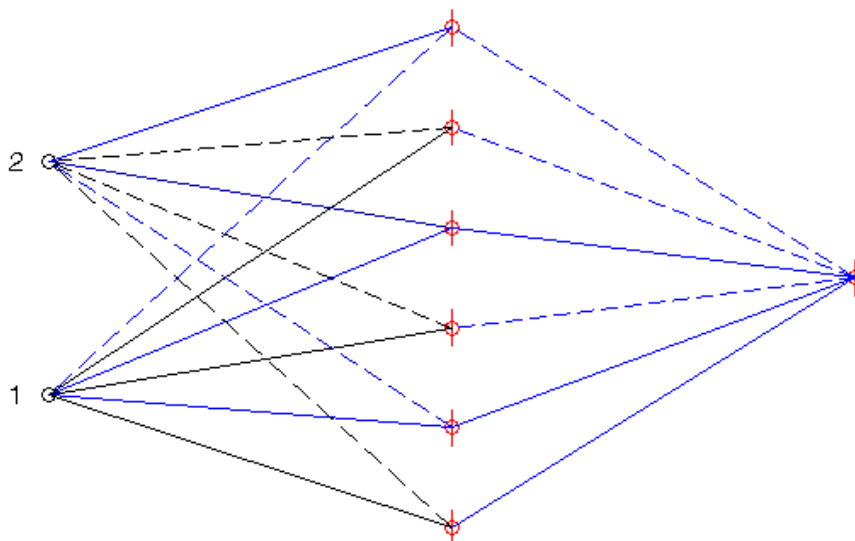
```

Network training started at 14. 9.49
iteration # 1   W = 9.962e+00iteration # 2   W = 5.950e+00iteration # 3   W = 5.755e+00iteration # 4   W = 5
Network training ended at 14. 9.50

```

Plot the structure of the NNARX model

```
figure, drawnet(W1,W2)
```



Extract the estimated parameters of the NNARX model

```

alfa=W2(1:end-1);
alfa_0=W2(end);
beta=W1(:,1:end-1);
beta_0=W1(:,end);

```

Compute the predicted output of the NNARX model using the validation dataset, defining the regressor vector phi as a column

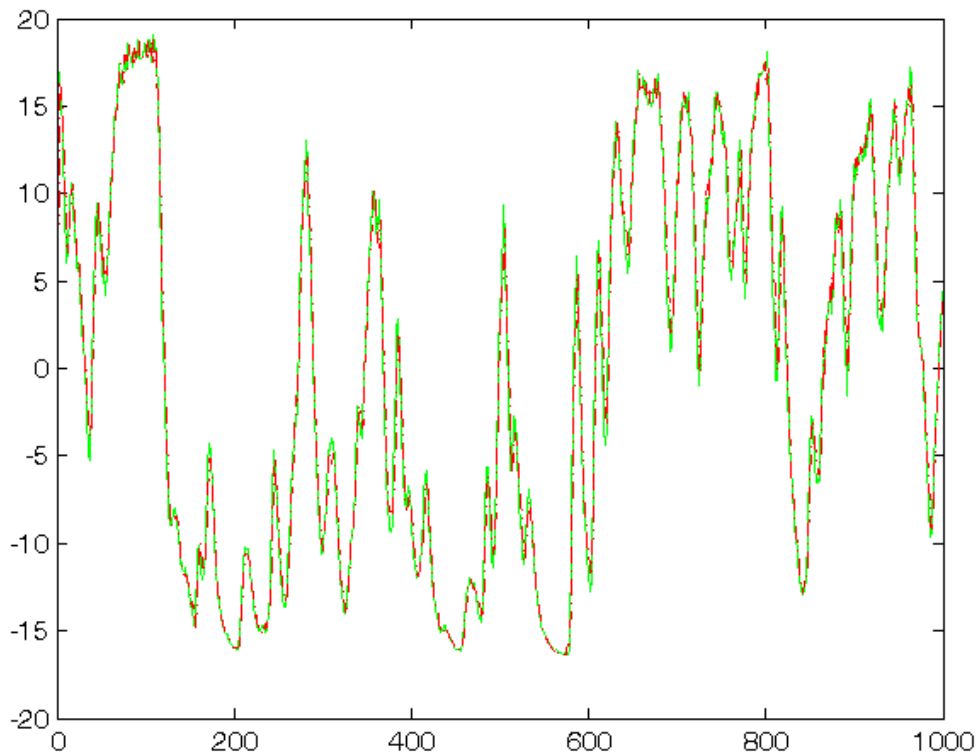
```

% Step 6: computation of predicted output

t_min=max([na+1,nb+nk]);
for t=t_min:Nv,
    phi= [yv_0mean(t-1:-1:t-na); uv_0mean(t-nk:-1:t-nk-nb+1)];
    yp(t)=alfa*tanh(beta*phi+beta_0)+alfa_0;
end

figure, plot(1:Nv,yv_0mean,'g', 1:Nv,yp,'r-.')

```



Compute the NNARX performances in terms of RMSE and MDL

```

% Step 7: computation of RMSE and MDL

N0=10; na,nb,nk
RMSE=sqrt(1/(Nv-N0))*norm(yv_0mean(N0+1:end)-yp(N0+1:end)')
n=na+nb;
MDL=n*log(Nv-N0)/(Nv-N0)+log(RMSE^2)

```

```

na =
    1
nb =
    1
nk =
    1
RMSE =
    1.0166
MDL =
    0.0469

```

Choosing the best model

The RMSE is minimized over 10 trials for each model (with $N_0=10$):

- $\text{nnarx}(1,1,1) \Rightarrow \text{RMSE} = 1.0128$
- $\text{nnarx}(1,1,2) \Rightarrow \text{RMSE} = 0.9444$
- $\text{nnarx}(1,1,3) \Rightarrow \text{RMSE} = 0.8250$
- $\text{nnarx}(1,1,4) \Rightarrow \text{RMSE} = 0.6590$
- $\text{nnarx}(1,1,5) \Rightarrow \text{RMSE} = 0.5705$
- $\text{nnarx}(2,2,1) \Rightarrow \text{RMSE} = 0.6907$
- $\text{nnarx}(2,2,2) \Rightarrow \text{RMSE} = 0.6414$
- $\text{nnarx}(2,2,3) \Rightarrow \text{RMSE} = 0.5448$
- $\text{nnarx}(2,2,4) \Rightarrow \text{RMSE} = 0.4894$
- $\text{nnarx}(2,2,5) \Rightarrow \text{RMSE} = 0.5205$
- $\text{nnarx}(3,3,1) \Rightarrow \text{RMSE} = 0.5983$
- $\text{nnarx}(3,3,2) \Rightarrow \text{RMSE} = 0.5197$
- $\text{nnarx}(3,3,3) \Rightarrow \text{RMSE} = 0.4757$
- $\text{nnarx}(3,3,4) \Rightarrow \text{RMSE} = 0.4548 \leq \text{best NNARX model}$
- $\text{nnarx}(3,3,5) \Rightarrow \text{RMSE} = 0.5160$
- $\text{nnarx}(5,5,1) \Rightarrow \text{RMSE} = 0.4801$
- $\text{nnarx}(5,5,4) \Rightarrow \text{RMSE} = 0.4600$
- $\text{nnarx}(7,7,1) \Rightarrow \text{RMSE} = 0.4666$
- $\text{nnarx}(7,7,4) \Rightarrow \text{RMSE} = 0.4557$
- $\text{nnarx}(9,9,1) \Rightarrow \text{RMSE} = 0.4678$
- $\text{nnarx}(9,9,4) \Rightarrow \text{RMSE} = 0.5459$